JPL PUBLICATION 82-100

(NASA-CR-169794) A COMPARISON OF FRAME SYNCHRONIZATION METHODS (Jet Propulsion Lab.) 23 p HC A02/MF A01 CSCL 17B

N83-16596

Unclas

G3/32 02602

A Comparison of Frame Synchronization Methods

Laif Swanson

December 15, 1982

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL PUBLICATION 82-100

A Comparison of Frame Synchronization Methods

Laif Swanson

December 15, 1982

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

ABSTRACT

Different methods are considered for frame synchronization of a concatenated block code/Viterbi link. Synchronization after Viterbi decoding, synchronization before Viterbi decoding based on hard-quantized channel symbols, and synchronization before Viterbi decoding based on soft-quantized channel symbols are all compared. For each scheme, the probability under certain conditions of true detection of sync within four 10,000-bit frames is shown in Table 1.

ACKNOWLEDGEMENT

The author wishes to acknowledge the technical support of Leslie J. Deutsch, Gary Lorden, Robert L. Miller, and Arthur L. Rubin, and helpful discussions with David J. Bell, Charles A. Greenhall, Steven D. Hall, Richard G. Lipes, and Robert J. McEliece in carrying out this research.

CONTENTS

Figures

Tables

SECTION I

INTRODUCTION

Deep space communications are usually encoded so that some channel errors can be corrected. Two types of codes are commonly used. One type is a block code, in which a fixed number of data bits is followed by a fixed number of "parity check" bits. The parity check bits add redundancy, so that some errors can be corrected. Examples of block codes are the Golay code, which has been used by Voyager, and the Reed-Solomon code, which may be used by Voyager for Uranus and Neptune encounters, and will be used by Galileo.

The other commonly used code is a rate 1/2, constraint length 7 convolutional code. In this code, each information bit is replaced by two channel symbols, each of which depends, in a different way, on this bit and the previous six bits.

In fact, many missions (including Voyager) use both coding systems simultaneously. Information bits on the spacecraft are encoded with a block code. Next this new string of bits is encoded with a convolutional code and transmitted. When the noisy, block-encoded, convolutionally-encoded message is received at the tracking station, the convolutional encoding must be undone, i.e., decoded. The process used in the Deep Space Network (DSN) is "maximum likelihood convolutional decoding," or "Viterbi decoding." After this process, there is a noisy version of the block-encoded data, which is decoded to reveal a noisy version of the original data. (The point of the processes is to have a less noisy version of the original data than would be had without the coding.)

To carry out the block decoding, the location where the blocks begin must be known. Determining where the blocks begin is the problem of "frame synchronization." This report describes a study into the effectiveness of various methods of frame synchronization.

SECTION II

METHODS OF FRAME SYNCHRONIZATION

In order to accomplish synchronization, there must be some flag that says "a block begins here." The flag need not be at the beginning of every block, if the lengths of blocks are known. In fact, sometimes constraints like interleaving make it impossible to flag the beginning of every block. Sometimes a certain amount of information is sent without block encoding, while another part of the information is block encoded. Of course, to avoid complete confusion, this must be done in an orderly fashion. One easy way to solve all these problems at once is to divide the data stream into "frames," each of which contains a certain number of blocks, and possibly other information, in a certain pattern. Then the problem reduces to identifying the beginning of the frame.

This is done by sending a certain string of symbols, called the "marker," at the beginning of every frame. Depending on the mission, the marker may or may not be block encoded, but in all cases the whole thing is convolutionally encoded. (See Figure 1).

The traditional method of frame synchronization is as follows. When the noisy, convolutionally-encoded, block encoded message is received, the convolutional code is decoded, resulting in a noisy, block-encoded message. The frame sync marker must be "found" now. When it is "found" (more about this later), the frame boundaries are known, and so block decoding is possible. This method is illustrated in Figure 2.

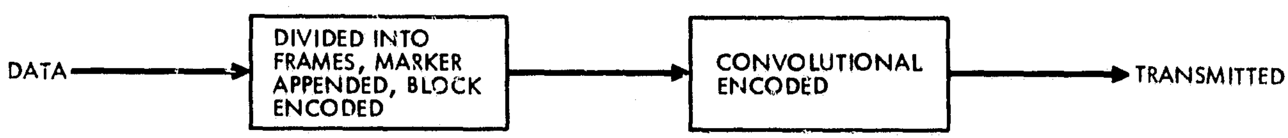


Figure 1. Coding Scheme on Spacecraft

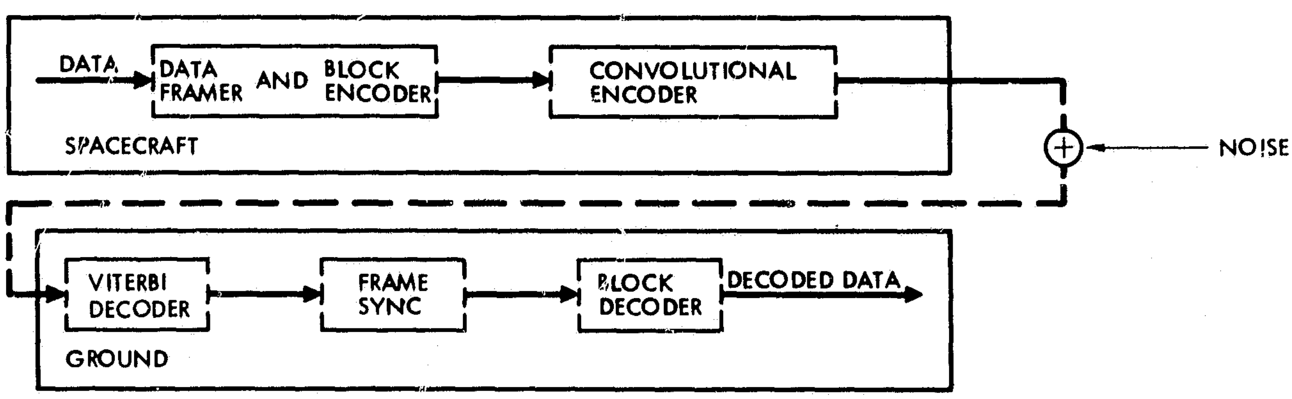


Figure 2. Conventional Concatenated Coding Scheme

Another method to be considered is to look for boundaries before Viterbi decoding. When the message is convolutionally encoded, the frame sync marker is convolutionally encoded as well, and so the convolutionally-encoded marker appears in the convolutionally encoded data stream. A frame synchronization device could therefore search before Viterbi decoding. This method is illustrated in Figure 3.

Both these methods are considered in this report. But one thing remains to be established, and that is what is meant by "finding" the marker. One possibility might be to say the marker is found when a string of L (L = the marker length) bits in the data stream agree exactly with the marker. Unfortunately, normal deep space communications signal-to-noise ratios make it fairly likely that the marker will contain some errors. (Exact numbers appear later.) So another possibility is to set a threshold T, and say that the marker is found when a string of bits disagrees with the marker in no more than T places. This is a good system when the frame is fairly short. In deep space communications, however, frames are typically about 10,000 bits long. Thus the probability that a string which disagrees with the marker in at most T places will appear in a random place in the frame is fairly large. (Again, numbers appear later.)

A third possibility is to look through a long string of bits -- as long as the frame -- and choose as the beginning of the frame the beginning of the L bits which look most like the marker. While this is an excellent system with moderate frame sizes [3], it has the same problem as the previous system in a case like the deep space communications: there is a substantial probability that a string which agrees with the marker in more places than the real (noisy) marker will appear.

A good compromise, allowing high enough threshold to cut down the probability of failure to acquire sync without raising the probability of incorrect acquisition of sync, is the threshold scheme with memory. In this scheme, a threshold T is chosen, and sync is "detected" each time L consecutive bits disagree with the marker in at most T places, but the search continues. Sync is <u>acquired</u> whenever sync is detected in the same place in two consecutive frames.

After sync is acquired, a system called "flywheeling" checks every frame to be sure that the marker is where it is expected. If for several consecutive frames the marker is not where it is expected, sync is lost and the acquisition procedure begins again.

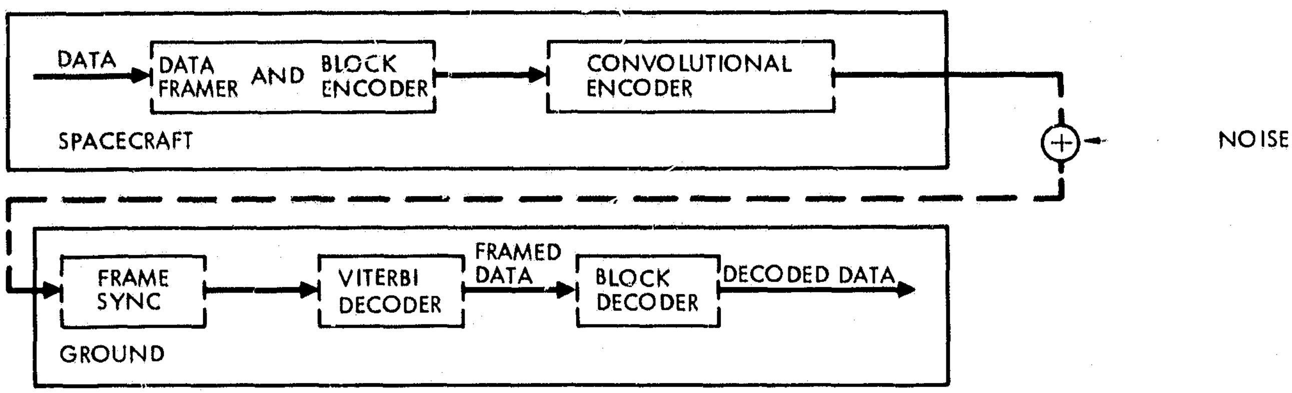


Figure 3. Alternate Concatenated Coding Scheme

SECTION III

CONCLUSIONS

Three threshold systems with memory were compared: search for frame sync marker after Viterbi decoding, search before Viterbi decoding based on hard-quantized channel symbols, and search before Viterbi decoding based on 3-bit quantized data symbols. Markers of length 24 bits and 32 bits were considered (except with the last system only 24-bit markers were considered). The method of search considered is a threshold scheme with memory. A threshold T is set, and sync is acquired when L (L is the length of the marker) bits (or channel symbols) disagree with the marker in at most T places in two consecutive frames.

For purposes of comparison, each method was studied at signal-to-noise ratio 2.3 dB (2.3 dB corresponds to Viterbi-decoded bit error rate 5×10^{-3}, the standard for imaging data), and thresholds were chosen to maximize the probability that sync is correctly acquired within four frames, each of length 10,000 bits, (a reasonable length for a sync buffer). These probabilities are shown in Table 1.

Table 1. Statistics for the Frame Sync Schemes

	Scheme 1 Synchronization After Viterbi Decoding	Scheme 2 Synchronization Before Viterbi Decoding (based on hard-quantized channel symbols)	Scheme 3 Synchronization Before Viterbi Decoding (based on 3-bit quantized channel symbols)
Probability of true acquisition within 4 frames at 2.3 dB with 24-bit marker	0.9993	0.9987	0.99996
Probability of true acquisition within 4 frames at 2.3 dB with 32-bit marker	0.9996	0.99996	

SECTION IV

FRAME SYNCHRONIZATION AFTER VITERBI DECODING

The frame synchronization method shown in Figure 2 is considered first. This is a tried-and-true method, currently used by Voyager.

An assumption made throughout is that a 10,000 bit frame is transmitted over a noisy Viterbi channel. The frame may be, in part or whole, block encoded, but that is not important in the analysis. The convolutional code is a rate 1/2, constraint length 7 code. Each frame begins with a marker of 24 or 32 bits. (Voyager uses a 32-bit marker; a 24-bit marker was originally proposed for the NASA End-to-End Data System (NEEDS).) Another assumption throughout is that a threshold system with memory is being used.

Two events are important to any study of frame synchronization. Because of channel errors, the marker can fail to be recognized at the beginning of the frame (failure to detect sync). Or the marker can appear in the data, causing a false detection of sync.

For a given threshold, the probability of failure to detect sync depends on the error patterns of the channel, but not on the marker. In random data alone (not overlapping the marker), the probability of false detection of sync depends only on the <u>length</u> of the marker. But the bits of the marker affect the probability of false detection of sync near (overlapping) the marker, sometimes critically. A marker is called "acceptable" [7] if, for every threshold, the probability of false detection of sync at each place near the marker is no larger than the probability of false detection of sync in data alone.

Because long frames are being considered, there are many places in which sync can be incorrectly acquired; very few of them overlap the marker and so, for the purposes of this report, a marker need not really be "acceptable." The major concern with these overlaps is that non-random fields, such as frame ID, which are near the marker could greatly increase the false alarm rate.

Since this section considers frame synchronization done after Viterbi decoding, Viterbi burst error statistics (described in [4]) are needed to determine the probability of failure to detect sync. This study assumes that the mean burst length is 13.3, that the average waiting time between bursts is 1516, and that the bit error rate within bursts is .585, the values for 2.3 dB.

The probabilities of failure to detect sync at threshold T (the probability that more than T bits in the marker are in error) are shown in Tables 2 and 3 for marker lengths 24 and 32, respectively. These probabilities are based on simulations of 32,000 frames using the geometric model.

Table 2. Probability of Failure to Detect 24-bit Sync Marker After Viterbi Decoding (Viterbi bit-error rate $5x10^{-3}$).

Threshold	Probability
0	0.024
1	0.020
2	0.0154
3	0.0120
4	0.0093
5	0.0070

Table 3. Probability of Failure to Detect 32-bit Sync Marker After Viterbi Decoding (Viterbi bit-error rate $5x10^{-3}$).

Threshold	Probability
0	0.0267
1	0.0217
2	0.0180
3	0.0153
4	0.0133
5	0.0112
6	0.0091
7	0.0074
8	0.0063

The probabilities of incorrect detection of sync at any given place can be calculated directly as

$$\sum_{K=0}^{T} \binom{L}{K} 2^{-L}$$

where L is marker length, based on the assumption that data in the frame, except the marker, is random. These probabilities appear in Tables 4 and 5.

Based on these probabilities, the probabilities of correct acquisition of sync for any acquisition scheme can be calculated immediately. This report will show those calculations for the scheme being studied (the scheme in which "sync

Table 4. Probability of False Detection of 24-bit Sync Marker, After Viterbi Decoding, in Random Data.

Threshold	Probability (per bit)
0	5.6×10^{-8}
1	1.5×10^{-6}
2	1.8×10^{-5}
3	1.4×10^{-4}
4	7.7×10^{-4}
5	3.3×10^{-3}

Table 5. Probability of False Detection of 32-bit Sync Marker, After Viterbi Decoding, in Random Data.

Threshold	Probability (per bit)
0	2.3×10^{-10}
1	7.7×10^{-9}
2	1.2×10^{-7}
3	1.3×10^{-6}
4	9.7×10^{-6}
5	5.7×10^{-5}
6	2.7×10^{-4}
7	1.1×10^{-3}
8	3.5×10^{-3}

is acquired when the marker is seen to threshold at the same place in two consecutive frames").

For a given marker length and threshold, let p be the probability of failure to detect sync (from Table 2 or 3), and let q be the probability of incorrect detection of sync per bit (Table 4 or 5). In order for sync to be acquired incorrectly, sync must be detected incorrectly twice in the same place, which happens with probability smaller than $10000q^2$. In order for sync to be declared correctly at the first possible time (by the time two markers have passed), it must be detected correctly twice, which happens with probability $(1-p)^2$, and sync must not be incorrectly declared before the correct marker is seen twice. Thus the probability of correct declaration of sync at the earliest possible time is at least $(1-p)^2 - 10{,}000q^2$. (The events of correct detection of sync in successive frames are not exactly independent, but are close enough for the numerical purposes of this report).

If sync is not acquired, incorrectly or correctly, by the end of two frames, it means (among other things) either that the marker was not correctly detected in the first frame, or that the marker was not detected correctly in the second frame. If the marker is detected in the two frames following the one in which it was missed, then sync will be acquired correctly. Each of these two events has probability $p(1-p)^2$, and so the probability of correct acquisition of sync within four frames (a typical size for a frame sync buffer) is at least $(1-p)^2 - 10{,}000q^2 + 2p(1-p)^2$. These numbers are tabulated in Tables 6 and 7.

Thus a marker of length 24, with threshold 2 or 3, would allow correct acquisition of sync within four frames with probability 0.9993. A marker of length 32, with threshold 5, would allow correct acquisition of sync within four frames with probability 0.9996.

Table 6. Probability of True Acquisition of Sync Within Four Frames, Using 24-bit Marker After Viterbi Decoding (at 5×10^{-3} bit-error rate).

Threshold	Probability
0	0.998
1	0.998
2	0.9993
3	0.9993
4	0.994
5	0.888

Table 7. Probability of True Acquisition of Sync Within Four Frames, Using 32-bit Marker After Viterbi Decoding (at 5×10^{-3} bit-error rate).

Threshold	Probability
0	0.9979
1	0.9986
2	0.9990
3	0.9993
4	0.9995
5	0.9996
6	0.9990
7	0.9877
8	0.8774

SECTION V

FRAME SYNCHRONIZATION BEFORE VITERBI DECODING

Frame synchronization can also be done before Viterbi decoding, based on convolutionally-encoded channel symbols, as shown in Figure 3. (This method is hard to implement in the case when frames are of different lengths, since information about where to look for the next marker is convolutionally encoded.) But in situations with constant frame length (those following NEEDS guidelines are examples), frame synchronization before Viterbi decoding is a possibility.

To calculate the performance of a system of this type, the same facts as before are needed: probability of seeing the (encoded) marker in encoded random data, and the probability of sufficient channel errors causing the synchronizer to fail to detect the marker at the beginning of a frame. This time, however, calculating these probabilities is a little different. Since these are convolutionally-encoded data instead of random symbols, the probability of false detection is a complicated number to calculate. With convolutionally-encoded data, unlike uncoded data, the probability of error also affects the probability of false detection. As part of this study, calculations were made to determine the probability of false detection of a 48 channel symbol sync word in convolutionally-encoded random data with error rate 0.1, with various thresholds. The numbers were then compared to the probability of seeing a 48 symbol word in random (not convolutionally-encoded) data. For thresholds of interest, these numbers were very close, as shown in Table 8. Because these numbers are so close, this report assumes throughout that the probability of false detection of a sync marker in convolutionally-encoded random data with error rate 0.1 is the same as the probability of false detection of a marker of the same length in random data.

Probability of failure to detect sync is calculated differently also, since in this case failure to detect is based not on Viterbi burst statistics, but on independent channel symbol errors.

In order to make the calculations, the length of the convolutionally encoded marker must be known. If a given sequence is embedded in a data stream to be convolutionally encoded, the first channel symbols to leave the encoder depend not only on the sequence but also on the bits already in the encoder's memory. For an encoder with memory length 6 (constraint length 7), the output of the encoder

Table 8. Probability of False Detection of a 48-Symbol Marker Using Various Thresholds in Convolutionally Encoded Data vs. Random Data with Symbol Error Probability 0.1.

Threshold	36	37	38	39	40
Convolutionally Encoded Data	3.58×10^{-4}	1.11×10^{-4}	3.06×10^{-5}	7.57×10^{-6}	1.68×10^{-6}
Random Data	3.59×10^{-4}	1.11×10^{-4}	3.08×10^{-5}	7.61×10^{-6}	1.65×10^{-6}

depends entirely on the given sequence only after the shift register has received the first six bits of the sequence, and is acting upon the seventh. Thus a marker of length L inserted in a data stream going into a (7, 1/2) encoder determines an encoded sequence of length 2(L-6). So this report will be considering 36 = 2(24-6) and 52 = 2(32-6) symbol markers.

If the convolutionally-encoded marker of length L channel symbols is detected whenever L channel symbols disagree with the convolutionally encoded marker in at most T places, then the probability of detecting the marker incorrectly in random data is

$$\sum_{K=L-T}^{L} \binom{L}{K} 2^{-L}$$

The probability that the convolutionally encoded marker is correctly detected is the same as the probability that at most T of the channel symbols in the marker are received incorrectly. This is

$$\sum_{K=0}^{T} \binom{L}{K} p^{K} (1-p)^{L-K}$$

where p=0.096 is the probability that a channel symbol is received incorrectly. Thus the probability that the encoded marker is not detected correctly is

$$1 - \sum_{K=0}^{T} \binom{L}{K} p^{K} (1-p)^{L-K} = \sum_{K=T+1}^{L} \binom{L}{K} p^{K} (1-p)^{L-K}$$

These probabilities are shown for L = 36 and L = 52, with various thresholds in Tables 9-12. Resulting probabilities of true acquisition of sync within four frames are shown in Tables 13 and 14. It may seem surprising that a 24-bit marker after Viterbi decoding performs slightly better than before decoding, while the opposite is true for a 32-bit marker. Six bits are lost in "overhead" when looking at the convolutionally encoded data; possibly the fact that these are a larger part of the 24-bit marker than of the 32-bit marker explains the difference in performance.

Before Viterbi decoding, however, there is more information than hard-quantized channel bits. Channel symbols are 12-bit quantized by the SSA, and the Viterbi decoder uses 3 bits. A frame synchronization scheme before Viterbi decoding can use as much of this information as it likes. Because of difficulty in computation using many quantization levels and long frame sync markers, this part of the study concentrated on three-bit quantization for a 36 channel symbol (24 bit) marker.

ORIGINAL PAGE IS
OF POOR QUALITY

Table 9. Probability of Failure to Detect 36 Channel Symbol Marker Before Viterbi Decoding (symbol error rate .096, hard-quantized symbols)

Threshold	Probability
0	0.9736
1	0.8725
2	0.6848
3	0.4588
4	0.2608
5	0.1263
6	0.0524
7	0.0188
8	0.0059
9	0.0016
10	0.0004
11	0.0001

Table 10. Probability of Failure to Detect 52 Channel Symbol Marker Before Viterbi Decoding (symbol error rate .096, hard-quantized symbols)

Threshold	Probability
0	0.9947
1	0.9657
2	0.8871
3	0.7480
4	0.5669
5	0.3824
6	0.2289
7	0.1218
8	0.0578
9	0.0246
10	0.0094
11	0.0032
12	0.0010
13	0.0003
14	0.0001
15	0.00001

Table 11. Probability of False Detection of 36-Symbol Sync Marker in Random Hard-Quantized Convolutionally Encoded Data

Threshold	Probability
0	1.5×10^{-11}
1	5.4×10^{-10}
2	9.7×10^{-9}
3	1.1×10^{-7}
4	9.7×10^{-7}
5	6.5×10^{-6}
6	3.5×10^{-5}
7	1.6×10^{-4}
8	6.0×10^{-4}
9	2.0×10^{-3}
10	5.7×10^{-3}
11	1.4×10^{-2}

Table 12. Probability of False Detection of 52-Symbol Sync Marker in Random Hard-Quantized Convolutionally Encoded Data

Threshold	Probability
0	2.2×10^{-16}
1	1.2×10^{-14}
2	3.1×10^{-13}
3	5.2×10^{-12}
4	6.5×10^{-11}
5	6.4×10^{-10}
6	5.2×10^{-9}
7	3.5×10^{-8}
8	2.0×10^{-7}
9	1.0×10^{-6}
10	4.5×10^{-6}
11	1.8×10^{-5}
12	6.4×10^{-5}
13	2.0×10^{-4}
14	6.0×10^{-4}
15	1.6×10^{-3}

ORIGINAL PAGE IS
OF POOR QUALITY

Table 13. Probability of True Acquisition of Sync Within Four Frames, Using 36 Channel Symbol Marker Before Viterbi Decoding at 0.096 Symbol Error Rate

Threshold	Probability
5	0.9521
6	0.9918
7	0.9987
8	0.9963
9	0.9600

Table 14. Probability of True Acquisition of Sync Within Four Frames, Using 52 Channel Symbol Marker Before Viterbi Decoding at 0.096 Symbol Error Rate

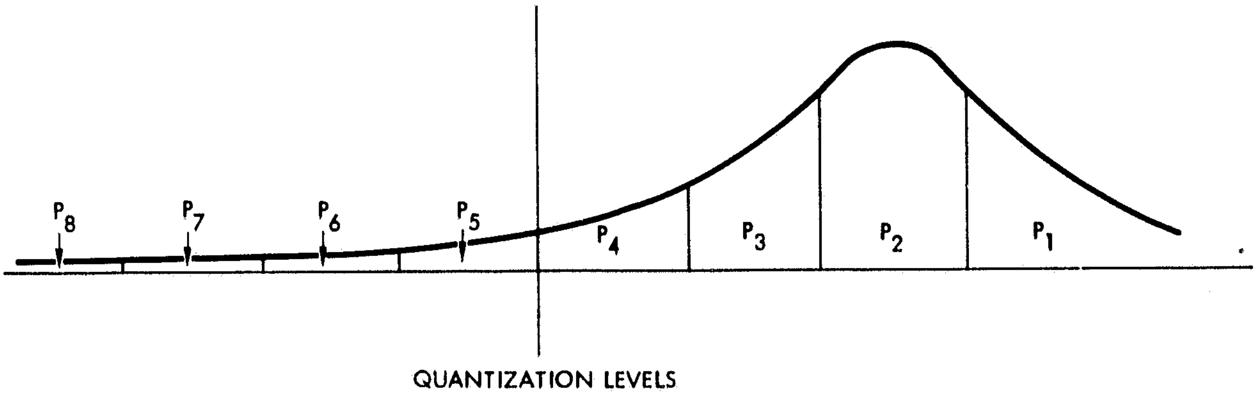
Threshold	Probability
9	0.9982
10	0.9997
11	0.99997
12	0.99996
13	0.9996

Of course, there are many more possible methods for detection of the marker in this system. Some counter scheme is necessary to weight each symbol according to degree of disagreement with a symbol in the frame sync marker. Statistical theory [2] shows that "log-likelihood" ratios should be used: to determine whether a sequence of 36 symbols is the marker, look at each symbol received, and calculate the logarithm for the ratio of the probability of receiving this in random data to the probability of receiving it if the marker was sent. These numbers should be added up for the thirty-six symbols in the candidate marker; the marker is assumed to be detected if this sum does not exceed a preassigned threshold. A much simpler counter scheme, however, presented in [3], works almost as well in this case, and is presented below.

Think of the eight (2^3) possible symbols received as four possible levels of agreement with the marker, and four possible levels of disagreement. To check whether the marker occurs at a certain place, check each received symbol against the respective marker symbol. If it agrees, the level of disagreement, and therefore the counter increment, is zero. If it disagrees, the counter increment is 1, 2, 3, or 4, depending on the level of disagreement. Add the 36 levels of disagreement together; sync is detected if this sum is no more than threshold.

ORIGINAL PAGE IS OF POOR QUALITY

At 2.3 dB, the probability of receiving each of these levels when a given symbol is sent are shown in Figure 4, derived from [8] and [9]. Using these, and the assumption that the probability of seeing the marker to threshold is the same as it would be with the symbols appearing independently, the probability of true acquisition of sync can be calculated. With threshold 15, the optimum for this, the probability of true acquisition of sync within four frames is 0.99996. The more complicated log-likelihood counter gives probability 0.99997.



$P_1 = 0.3971$, $P_2 = 0.2059$, $P_3 = 0.1802$, $P_4 = 0.1209$,
$P_5 = 0.0621$, $P_6 = 0.0244$, $P_7 = 0.0074$, $P_8 = 0.0020$

Figure 4. Probabilities of each Quantization Level at Symbol Error Rate 0.096

SECTION VI

CHOICE OF MARKERS

Markers should be chosen to minimize agreements between initial and terminal segments, in order to minimize probability of incorrect detection of sync at places overlapping the marker. Whether relevant agreements are in the uncoded marker or in the convolutionally-encoded version of the marker depends on which method of sync acquisition is being used. Some markers are usable in both contexts. For example, the current NEEDS marker is 6650FB65 hex. While neither this word nor its convolutionally-encoded version meets the definition of "acceptable," both are fairly close to it; of all possible extensions of the original NEEDS sync word, 6650FB, this is the only one to offer such low overlap agreements in both uncoded and convolutionally-encoded versions. (The original 24-bit NEEDS sync word was extended because of the differences in performance between a 24-bit marker and a 32-bit marker.)

SECTION VII

REVIEW

The performance of 24-bit and 32-bit markers to synchronize 10,000 bit frames at 2.3 dB was analyzed, comparing synchronization before and after Viterbi decoding. In all cases, the probability of true acquisition of sync within four frames was calculated. Of the methods studied, synchronization before Viterbi decoding based on soft-quantized channel symbols performed best. For the longer frame-sync marker, synchronization before Viterbi decoding based on hard-quantized channel symbols is better than frame synchronization after Viterbi decoding, while for the shorter marker, the opposite is true. All these probabilities were presented in Table 1.

REFERENCES

1. K. Y. Liu and J. J. Lee, "An Experimental Study of the Concatenated Reed-Solomon/Viterbi Channel Coding System Performance and its Impact on Space Communications," Jet Propulsion Laboratory Publication 81-58.

2. G. Lorden, "Procedures for Reacting to a Change in Distribution," The Annals of Mathematical Statistics, 1971, Vol. 42, No. 6, 1897-1908.

3. J. L. Massey, "Optimum Frame Synchronization," IEEE Transactions on Communications, Vol. Com-20, No. 2, April 1972, 115-119.

4. R. L. Miller, L. J. Deutsch, and S. A. Butman, "On the Error Statistics of Viterbi Decoding and the Performance of Concatenated Codes," September 1, 1981, Jet Propulsion Laboratory Publication 81-9.

5. J. P. Odenwalder, "Optimum Decoding of Convolutional Codes," Ph.D dissertation, System Science Department, University of California, Los Angeles, 1970.

6. J. P. Odenwalder, et al. (Linkabit Corp.), "Hybrid Coding Systems Study Final Report," September 1972, NASA-CR-114486.

7. R. A. Scholtz, "Frame Synchronization Techniques," IEEE Transactions on Communications, Vol. Com-28, August 1980, No. 8, 1204-1213.

8. Symbol Synchronizer Assembly, Jet Propulsion Laboratory Technical Manual TM 01106 (internal document).

9. Instruction Manual, Maximum Likelihood Convolutional Decoder, Jet Propulsion Laboratory TM 14001 (internal document).